**AZURE CICD PIPELINE**



**Summary of Azure CI/CD Pipeline**

The Azure CI/CD pipeline automates the build, test, and deployment of a multi-microservice application (Python, .NET, Node.js) using **Azure DevOps**, **ACR**, **AKS**, and **GitOps** (Argo CD). Here’s the workflow:

**Architecture Overview**

1. **Components**:
   * **Azure Repos**: Hosts source code for microservices (voting app, worker, results).
   * **CI Pipeline**: Builds Docker images for each microservice and pushes them to **Azure Container Registry (ACR)**.
   * **AKS Cluster**: Hosts the deployed application.
   * **GitOps (Argo CD)**: Monitors the Git repository for Kubernetes manifest changes and auto-deploys updates to AKS.
2. **Workflow**:
   * **Continuous Integration (CI)**:
     + Triggered by code commits to Azure Repos.
     + Builds Docker images for each microservice (Python, .NET, Node.js).
     + Pushes images to ACR with dynamic tags (e.g., voting-app:65).
   * **Continuous Delivery (CD)**:
     + **Update Stage**: A shell script updates Kubernetes manifests in Git with the new image tag.
     + **Argo CD**: Detects Git changes and deploys the updated manifests to AKS.
3. **Key Tools**:
   * **Azure Pipelines**: For CI stages (build, push, update).
   * **ACR**: Stores Docker images.
   * **AKS**: Kubernetes cluster for deployment.
   * **Argo CD**: GitOps tool for declarative Kubernetes deployments.

4. **Azure Repos → CI Pipeline → ACR → GitOps (Argo CD) → AKS.**